

Alternating Optimisation and Quadrature for Robust Reinforcement Learning

Supratik Paul

Department of Computer Science
University of Oxford
supratik.paul@cs.ox.ac.uk

Michael A. Osborne

Department of Engineering Science
University of Oxford
mosb@robots.ox.ac.uk

Kamil Ciosek

Department of Computer Science
University of Oxford
kamil.ciosek@cs.ox.ac.uk

Shimon Whiteson

Department of Computer Science
University of Oxford
shimon.whiteson@cs.ox.ac.uk

Abstract

Bayesian optimisation has been successfully applied to a variety of reinforcement learning problems. However, the traditional approach for learning optimal policies in simulators does not utilise the opportunity to improve learning by adjusting certain *environment variables* – state features that are randomly determined by the environment in a physical setting but are controllable in a simulator. This paper considers the problem of finding an optimal policy while taking into account the impact of environment variables. We present *alternating optimisation and quadrature* (ALOQ), which uses Bayesian optimisation and Bayesian quadrature to address such settings. ALOQ is robust to the presence of significant rare events, which may not be observable under random sampling, but have a considerable impact on determining the optimal policy. We provide experimental results demonstrating our approach learning more efficiently than existing methods.

Introduction

A key consideration when applying *reinforcement learning* (RL) to a physical setting is the risk and expense of running trials, for example while learning the optimal policy for a robot. To address this, learning is often conducted in simulators. Although this is cheaper and safer than running physical trials, the computational cost of conducting each simulated trial can still be quite high. The challenge then is to develop algorithms that are sample efficient, i.e., that minimise the number of such trials. In such settings, *Bayesian optimisation* (BO) (Brochu, Cora, and de Freitas 2010) is an appealing approach because it is highly sample efficient and has been successfully applied to RL in multiple domains (Lizotte et al. 2007; Martinez-Cantin et al. 2007; Martinez-Cantin et al. 2009; Cully et al. 2015; Calandra et al. 2015).

However, the traditional approach to BO does not take advantage of an opportunity afforded by simulators: the ability to adjust certain *environment variables*, state features that cannot be controlled in a physical setting but are (randomly) determined by the environment. For example, when learning to fly a helicopter under different wind conditions (Koppejan and Whiteson 2011), we typically cannot control the wind in physical trials but can easily do so in a simulator.

A naïve approach would be to simply randomly sample values for these environment variables in each trial, so as to

estimate expected performance. However, this approach is not robust to *significant rare events* (SREs), i.e., it fails any time there are rare events that substantially affect expected performance. For example, some rare wind conditions may increase the risk of crashing the helicopter. Since crashes are so catastrophic, avoiding them is key to maximising expected performance, even though the wind conditions contributing to the crash occur only rarely. In such cases, the naïve approach will not see such rare events often enough to learn an appropriate response.

In this paper, we propose a new approach called *alternating optimisation and quadrature* (ALOQ) that is specifically aimed towards learning policies that are robust to these rare events. The main idea is to construct a *Gaussian process* (GP) that models performance as a function of both the policy and the environment variables and then, at each timestep, to use BO and *Bayesian quadrature* (BQ) in turn to select a policy and environment setting, respectively, to evaluate.

We apply ALOQ to a number of test problems, as well as robot control problems. Our results demonstrate that ALOQ learns better and faster than multiple baselines, including a naïve approach, ALOQ with certain components ablated, and an existing GP-based method for coping with environment variables. Finally, we show how ALOQ can help cope with model uncertainty in model-based RL.

Related Work

Frank, Mannor, and Precup (2008) also consider the problems posed by SREs in RL. In particular, they propose an approach based on importance sampling for efficiently evaluating policies whose expected value may be substantially affected by rare events. While their approach is based on *temporal difference* (TD) methods, we take a BO-based policy search approach. Unlike TD methods, BO is well suited to settings in which sample efficiency is paramount and/or where assumptions (e.g., the Markov property) that underlie TD methods cannot be verified. BO has had empirical success in such settings (Lizotte et al. 2007; Martinez-Cantin et al. 2007; Martinez-Cantin et al. 2009; Cully et al. 2015; Calandra et al. 2015).

More importantly, Frank, Mannor, and Precup assume prior knowledge of the SREs, such that they can directly alter the probability of such events during policy evaluation. By contrast, a key strength of ALOQ is that it requires

only that a set of environment variables can be controlled in the simulator, without assuming any prior knowledge about whether SREs exist, or about the settings of the environment variables that might trigger them.

Williams, Santner, and Notz (2000) consider a problem setting they call the *design of computer experiments* which is essentially identical to our setting. They also propose an approach that alternates between BO and BQ. However, their approach, which we discuss further in the method section, is applicable only to discrete environment variables whereas ALOQ can handle both discrete and continuous ones. Furthermore, our experiments show that ALOQ is faster computationally; more robust to SREs; and, unlike the approach of Williams et al., outperforms a baseline that randomly samples the environment variable.

Finally, Krause and Ong (2011) also consider the problem of optimising performance in the presence of environment variables. However, whereas in our setting we optimise performance after marginalising out the environment variable, they address a contextual bandit setting in which the learned policy conditions on the environment variable. This renders their approach inapplicable to the settings we consider with unobservable environmental variables.

Background

Given a (possibly noisy) black-box objective function $f(\mathbf{x}) : \mathbb{R}^d \rightarrow \mathbb{R}$, we assume that it has been drawn from a GP prior. A GP (Rasmussen and Williams 2005) is completely specified by a mean function $m(\mathbf{x})$ and a (positive definite) covariance function $k(\mathbf{x}, \mathbf{x}')$ and has the property that any finite set of points can be expressed as a multivariate Gaussian distribution. The distribution of any point, given a set of observed points, can be expressed in closed form due to the conditional and marginalisation properties of the Gaussian.

The prior mean function of the GP is often assumed to be 0 for convenience. A popular choice for the covariance function is the class of stationary functions of the form $k(\mathbf{x}, \mathbf{x}') = k(\mathbf{x} - \mathbf{x}')$, which may have some hyperparameters ζ . Observed data points $\mathcal{D} = \{\mathbf{x}_i, f(\mathbf{x}_i)\}_{i=1}^N$ are used to update the posterior belief about the objective function through the posterior distribution $p(f | \mathcal{D}, \zeta) \propto p(f | \zeta)p(\mathcal{D} | f, \zeta)$. We follow a full Bayesian approach and compute the marginalised posterior distribution $p(f | \mathcal{D})$ by first placing a hyperprior distribution on ζ and then marginalising it out from $p(f | \mathcal{D}, \zeta)$. In practice, an analytical solution for this is unlikely to exist so we estimate $\int p(f | \mathcal{D}, \zeta)p(\zeta | \mathcal{D})d\zeta$ using *Monte Carlo quadrature*, described next. For ease of notation, we drop ζ from the conditioning set in the rest of the paper.

Monte Carlo quadrature estimates integrals of the form $\bar{f} = \int f(\mathbf{x})p(\mathbf{x})d\mathbf{x}$, where $p(\mathbf{x})$ is the probability density of \mathbf{x} . We simply sample $(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N)$ from $p(\mathbf{x})$ and estimate the integral as $\bar{f} \approx \frac{1}{N} \sum_{i=1}^N f(\mathbf{x}_i)$. N needs to be sufficiently large to ensure an accurate estimate of the integral, and hence this method should only be used if f is cheap to evaluate.

BQ (O'Hagan 1991; Rasmussen and Ghahramani 2003) is an alternative to Monte Carlo quadrature for settings where

f is computationally expensive to evaluate. BQ begins by taking a GP prior for $f(\mathbf{x})$, and then uses evaluations of the integrand (at selected nodes) to compute a posterior for f . This posterior can then be used to compute a univariate Gaussian posterior for the integral, \bar{f} , whose mean and variance can be computed analytically for particular choices¹ of GP covariance and prior $p(\mathbf{x})$. If no analytical solution exists, we can approximate the mean and variance via Monte Carlo quadrature by sampling from the posterior of f .

BO addresses the problem of optimising $f(\mathbf{x})$ within some compact set \mathcal{A} , i.e. finding \mathbf{x}^* :

$$\mathbf{x}^* = \operatorname{argmax}_{\mathbf{x} \in \mathcal{A} \subset \mathbb{R}^d} f(\mathbf{x}). \quad (1)$$

BO works by selecting for evaluation at each timestep the \mathbf{x} that maximises an *acquisition function* such as *expected improvement* (EI) (Moćkus 1975; Jones, Schonlau, and Welch 1998) or *upper confidence bound* (UCB) (Cox and John 1992; Cox and John 1997). Defining \mathbf{x}^+ as the current optimal evaluation, i.e., $\mathbf{x}^+ = \operatorname{argmax}_{\mathbf{x}_i} f(\mathbf{x}_i)$, EI seeks to maximise the expected improvement over the current optimum:

$$\alpha_{EI}(\mathbf{x}) = \mathbb{E}[I(\mathbf{x})], \quad I(\mathbf{x}) = \max\{0, f(\mathbf{x}) - f(\mathbf{x}^+)\}. \quad (2)$$

By contrast, UCB does not depend on \mathbf{x}^+ but still acknowledges the uncertainty in our estimate while computing the potential for improvement at any point:

$$\alpha_{UCB}(\mathbf{x}) = \mu(\mathbf{x}) + \kappa\sigma(\mathbf{x}), \quad (3)$$

where μ and σ^2 are the mean and variance, and κ controls the exploration-exploitation tradeoff.

Problem Setting & Method

We assume access to a computationally expensive simulator that takes as input a policy $\pi \in \mathcal{A}$ and environment variable $\theta \in \mathcal{B}$ and produces as output $f(\pi, \theta) \in \mathcal{R}$, where both \mathcal{A} and \mathcal{B} belong to some compact sets in \mathbb{R}^{d_π} and \mathbb{R}^{d_θ} , respectively. We also assume that $p(\theta)$, the probability distribution over θ , is known. Defining $f_i = f(\pi_i, \theta_i)$, we assume an initial dataset $\mathcal{D}_{1:l} = \{(\pi_1, \theta_1, f_1), (\pi_2, \theta_2, f_2), \dots, (\pi_l, \theta_l, f_l)\}$. Our objective is to find an optimal policy π^* :

$$\pi^* = \operatorname{argmax}_{\pi} \bar{f}(\pi) = \operatorname{argmax}_{\pi} \mathbb{E}_{\theta} [f(\pi, \theta)]. \quad (4)$$

Let us first consider a naïve approach that disregards θ , applies BO directly to $\tilde{f}(\pi) = f(\pi, \theta)$, with only one input π , and attempts to estimate π^* . Formally, this approach models \tilde{f} as a GP with a zero mean function and a suitable covariance function $k(\pi, \pi')$. For any given π , the variation in f due to different settings of θ is treated as observation noise. To estimate π^* , the naïve approach applies BO, while sampling θ from $p(\theta)$ at each timestep. This approach will

¹Most notably, the posterior for the integral is closed form for combinations of a squared exponential covariance function and a prior that is Gaussian or a mixture of Gaussians. Many other combinations of covariance and prior also yield closed-form posteriors (Briol et al. 2015).

almost surely fail, as it is unlikely to sample SREs often enough to learn a suitable response.

A better approach is to model $f(\pi, \theta)$, acknowledging both its inputs, as a GP: $f \sim GP(m, k)$. At timestep $t + 1$, we then simultaneously select π_{t+1} and θ_{t+1} : $(\pi_{t+1}, \theta_{t+1}) = \operatorname{argmax}_{\pi, \theta} \alpha_{SO}(\pi, \theta)$. Here α_{SO} is a novel acquisition function, based on EI, that measures the expected improvement from the given (π, θ) , while marginalising out θ :

$$\alpha_{SO}(\pi, \theta) = \mathbb{E}_{f(\pi, \theta) | \mathcal{D}_{1:t}} \left[\max\{0, \delta_{SO}\} \right],$$

where $\delta_{SO} = \bar{f}(\pi) | \mathcal{D}_{1:t+1} - \bar{f}(\pi^+) | \mathcal{D}_{1:t}$. (5)

Here π^+ is the best π found up until timestep t . Unlike (2), the max operator together with a stochastic $\bar{f}(\pi^+) | \mathcal{D}_{1:t}$ makes $\alpha_{SO}(\pi, \theta)$ analytically intractable. We could approximate it using Monte Carlo sampling but the computational cost, increasing with t , makes this impractical.

Instead of simultaneously selecting π and θ , another approach is to select them in alternating fashion: first select π using a BO acquisition function on $\bar{f}(\pi) | \mathcal{D}$, then select θ using a quadrature acquisition function that conditions on the selected π . This is the approach taken by Williams, Santner, and Notz (2000). The BO acquisition function of their algorithm, which we refer to as WSN, is based on EI:

$$\alpha_{WSN}(\pi) = \mathbb{E}_{\bar{f}(\pi) | \mathcal{D}_{1:t}} \left[I(\pi) | \mathcal{D}_{1:t} \right]$$

where $I(\pi) = \max\{0, \bar{f}(\pi) - \bar{f}(\pi^+)\}$. (6)

α_{WSN} cannot be computed analytically since $\bar{f}(\pi^+) | \mathcal{D}_{1:t}$ is a random variable. This is addressed by applying the identity $\mathbb{E}[I(\pi) | \mathcal{D}_{1:t}] = \mathbb{E}\{\mathbb{E}_{\bar{f}(\pi^+) | \mathcal{D}_{1:t}}[I(\pi) | \mathcal{D}_{1:t}, \bar{f}(\pi^+)]\}$ and using Monte Carlo sampling to approximate the inner expectation. This requires computing the joint distribution of $\{\bar{f}(\pi_1), \bar{f}(\pi_2), \dots, \bar{f}(\pi_t) | \mathcal{D}_{1:t}\}$, which in turn requires performing predictions on $t \times N_\theta$ points, i.e., all the t observed π 's paired with each of the N_θ support points for the environment variable. This is prohibitively expensive in practice, even for moderate t , as the computational complexity of GP predictions scales quadratically with the number of predictions. Moreover, as we highlight later in this section, WSN with a stationary covariance function, as the authors propose, is unsuited to modelling SREs as it cannot capture the different length scales that characterise such events. Finally, the formulation of WSN is such that it can only be applied to settings where θ is discrete and $\bar{f}(\pi) | \mathcal{D} = \sum_{\theta} p(\theta) \{f(\pi, \theta) | \mathcal{D}\}$.

We propose a new, simpler, and more effective alternating scheme called *alternating optimisation and quadrature* (ALOQ). Without loss of generality, we assume that the prior mean function m is 0 and use a suitable stationary kernel as the covariance function k . Our estimate of π^* is thus:

$$\hat{\pi}^* = \operatorname{argmax}_{\pi} \mathbb{E}_{\bar{f}(\pi) | \mathcal{D}_{1:t}} [\bar{f}(\pi)]. \quad (7)$$

Unlike WSN, we make no restrictive assumptions about θ . While for discrete θ the estimate for $\bar{f}(\pi) | \mathcal{D}$ is straightforward, for continuous θ we apply Monte Carlo quadrature. Although this requires sampling a large number of θ and

evaluating the corresponding $f(\pi, \theta) | \mathcal{D}$, it is feasible since we evaluate $f(\pi, \theta) | \mathcal{D}$ not from the expensive simulator, but from the computationally cheaper GP.

As noted earlier, using the EI acquisition function is not practical for ALOQ. However, the UCB acquisition function is a natural choice since it bypasses the challenges with estimating $\bar{f}(\pi^+) | \mathcal{D}$ at each timestep. Our acquisition function for π is thus:

$$\alpha_{ALOQ}(\pi) = \mu(\bar{f}(\pi) | \mathcal{D}) + \kappa \sigma(\bar{f}(\pi) | \mathcal{D}), \quad (8)$$

and at timestep $t + 1$ we set $\pi_{t+1} = \operatorname{argmax}_{\pi} \alpha_{ALOQ}(\pi) | \mathcal{D}_{1:t}$. Once π_{t+1} has been selected, we follow Osborne et al. (2012) and seek to minimise the posterior variance of $\bar{f}(\pi_{t+1})$ to select θ_{t+1} :

$$\theta_{t+1} | \pi_{t+1} = \operatorname{argmin}_{\theta} \mathbb{V}(\bar{f}(\pi_{t+1}) | \mathcal{D}_{1:t}, \pi_{t+1}, \theta). \quad (9)$$

Although the approach described so far actively selects π and θ through BO and BQ, it is unlikely to perform well in practice. A key observation is that the presence of SREs, which we hope ALOQ will address, implies that the scale of f varies considerably, e.g., from dangerous wind conditions to normal ones. This nonstationarity cannot be modelled with our stationary kernel. Therefore, we must transform the inputs to ensure stationarity of f . In particular, we transform the inputs along both π and θ using Beta CDFs with parameters (α, β) , as in Snoek et al. (2013).

While the resulting algorithm should be able to cope with SREs, the $\hat{\pi}^*$ that it returns at each iteration may still be poor, since our BQ evaluation of $\bar{f}(\pi)$ leads to a noisy approximation of the true objective function. In low-dimensional settings, adequate coverage of the response surface can be obtained using few data points. However, in higher dimensional problems, *intensification* (Bartz-Beielstein, Lasarczyk, and Preuss 2005; Hutter et al. 2009), i.e., re-evaluation of selected policies, is essential. Therefore, ALOQ performs two function evaluations at each timestep. In the first evaluation, (π, θ) is selected via the BO/BQ scheme described above. In the second stage, $(\hat{\pi}^*, \theta^*)$ is evaluated, where $\hat{\pi}^* \in \pi_{1:n}$ and $\theta^* | \hat{\pi}^*$ is selected using the BQ acquisition function.

Algorithm 1 summarises ALOQ. ζ denotes the combined set of hyperparameters of k and the parameters of the Beta CDFs. The corresponding hyperpriors are denoted by $p(\zeta)$.

Experimental Results

To evaluate ALOQ, we applied it to three types of problems: 1) artificial test functions, including those used by Williams, Santner, and Notz (2000), 2) a simulated robot arm control task, and 3) a variation of the latter that considers model uncertainty.

For each problem, we compare ALOQ to several baselines: 1) the *naïve* method described in the previous section; 2) the WSN method; 3) *random quadrature*, which is like ALOQ but samples θ randomly from $p(\theta)$ instead of choosing it actively; 4) *unwarped ALOQ*, which does not perform Beta warping of the inputs; and 5) *one-step ALOQ*, which does not use intensification. All plotted results are the median of 20 independent runs. We use the same parameters

Algorithm 1 ALOQ

input A black box function $f(\pi, \theta)$, initial dataset $\mathcal{D}_{1:l}$, the number of function evaluations L , GP prior with hyperparameters ζ and corresponding hyperpriors $p(\zeta)$.

- 1: **for** $n = l + 1, l + 3, \dots, L - 1$ **do**
 - 2: Draw a random sample $\{\zeta_1, \zeta_2, \dots, \zeta_z\}$ from $p(\zeta | \mathcal{D}_{1:n-1})$ using a Monte Carlo method
 - 3: Compute the marginalised posterior distribution:

$$p(f | \mathcal{D}_{1:n-1}) = \int p(f | \mathcal{D}_{1:n-1}, \zeta) p(\zeta | \mathcal{D}_{1:n-1}) d\zeta$$

$$\approx \frac{1}{z} \sum_{i=1}^z p(f | \mathcal{D}_{1:n-1}, \zeta_i)$$
 - 4: Use Monte Carlo quadrature to estimate $p(\bar{f} | \mathcal{D}_{1:n-1})$
 - 5: Use the BO acquisition function to select $\pi_n = \operatorname{argmax}_{\pi} \alpha_{ALOQ}(\bar{f}(\pi) | \mathcal{D}_{1:n-1})$
 - 6: Use the BQ acquisition function to select $\theta_n | \pi_n = \operatorname{argmin}_{\theta} \sigma^2(\bar{f}(\pi_n) | \mathcal{D}_{1:n-1}, \pi_n, \theta)$
 - 7: Evaluate $f_n = f(\pi_n, \theta_n)$ and update $\mathcal{D}_{1:n-1}$ to $\mathcal{D}_{1:n}$
 - 8: Find $\hat{\pi}^* = \operatorname{argmax}_{\pi_i} f(\pi_i) | \mathcal{D}_{1:n}$ and $\theta^* | \hat{\pi}^*$ using the BQ acquisition function.
 - 9: Evaluate $f_{n+1} = f(\hat{\pi}^*, \theta^*)$ and update $\mathcal{D}_{1:n}$ to $\mathcal{D}_{1:n+1}$
 - 10: **end for**
- output** $\pi^* = \operatorname{argmax}_{\pi_i} \bar{f}(\pi_i) | \mathcal{D}_{1:L} \quad i = 1, 2, \dots, L$
-

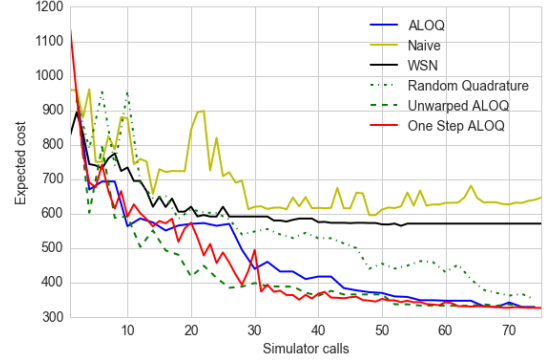
and hyperpriors for all algorithms across each of the three sets of experiments. Additional details about the experimental setup as well as additional experiments can be found in the supplementary material.

Artificial Test Functions

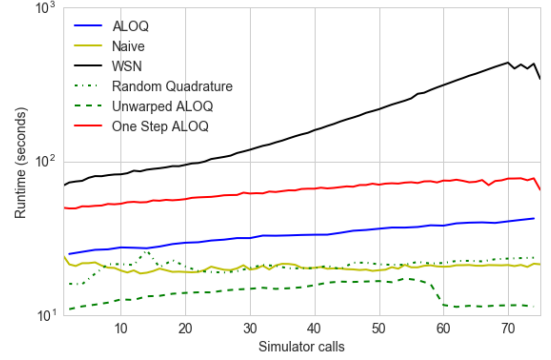
We begin with the *Branin* test function used by Williams, Santner, and Notz (2000) to evaluate WSN. This is a four-dimensional problem, with two dimensions treated as discrete environmental variables with a total of 12 support points. Figure 1a plots $\mathbb{E}_{\theta}[\hat{\pi}^*, \theta]$ for the $\hat{\pi}^*$ estimated by each algorithm at each timestep. ALOQ, random quadrature, unwarped ALOQ, and one-step ALOQ all substantially outperform WSN.

Figure 1b plots the per-step runtime of each algorithm, i.e., the time taken to process one data point. WSN takes significantly longer than ALOQ or the other baselines, and shows a clear increasing trend. The reduction in time near the end is a computational artefact due to resources being freed up as some runs finish faster than others. See the supplementary material for similar results on *Hartmann-6*, another function tested by Williams, Santner, and Notz.

The slow runtime of WSN is as expected (see previous section). However, its failure to outperform random quadrature is surprising, as these are the test problems Williams, Santner, and Notz (2000) use in their own evaluation. However, they never compared WSN to these (or any other) baselines. Consequently, they never validated the benefit of modelling θ explicitly, much less selecting it actively. In retrospect, these results make sense because the function is not characterised by significant rare events and there is no other



(a) Branin (min) - expected value of $\hat{\pi}^*$



(b) Branin (min) - runtime comparison

Figure 1: Comparison of performance and runtime of all methods on the Branin test function used by WSN.

a priori reason to predict that simpler methods will fail.

Hence, these results underscore the fact that a meaningful evaluation must include a problem with SREs, as such problems do demand more robust methods. To create such an evaluation, we formulated two test functions, F-SRE1 and F-SRE2, that are characterised by significant rare events. We present the results of F-SRE1 here; see the supplementary material for the F-SRE2 results. Figure 2a shows the contour plots of FSRE-1, which has a narrow band of θ in which the scale of the rewards is much larger (to make the plots more readable, we downscaled the region corresponding to the SRE, $\theta < 0$, by a factor of 10).

Figures 2b, which plots the performance of all methods on FSRE-1, shows that ALOQ substantially outperforms all the other algorithms except for one-step ALOQ. As expected, intensification does not yield any additional benefit in this low-dimensional problem. However, our experiments presented next show that intensification is crucial for success in higher dimensional problems.

The final learned policy, i.e., $\hat{\pi}^*$, of each algorithm is shown as a vertical line in Figure 2a, along with π^* (the true maximum). These lines illustrate that properly accounting for significant rare events can lead to learning qualitatively different policies. Figure 2c shows in log-scale the per-step runtime of each algorithm. As before, WSN is far slower than the other methods.

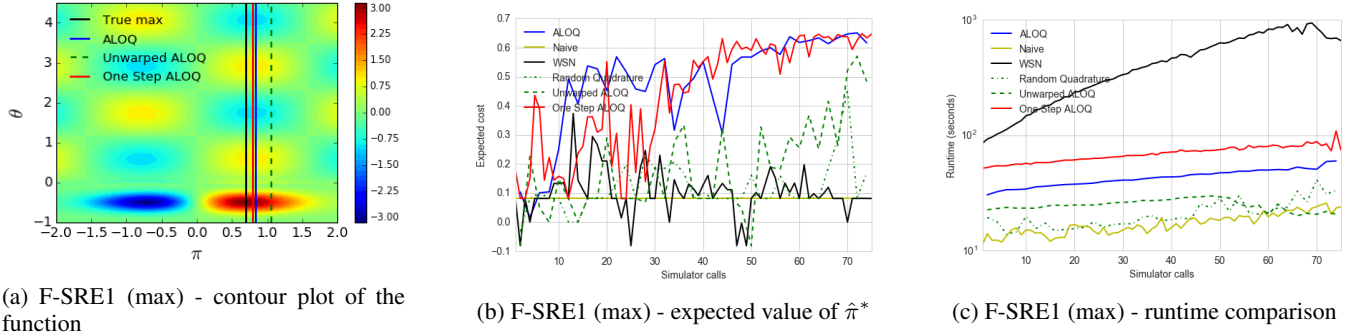


Figure 2: Contour plot of F-SRE1 (values in SRE region - $\theta < 0$, has been reduced by a factor of 10) as well as performance and runtime (note the log scale) of all methods on these problems.

Robotic Arm Simulator

Next, we evaluated ALOQ’s performance on a robot control problem implemented in a kinematic simulator. Details of the setup can be found in the supplementary material. The goal is to configure each of the three controllable joints of a robot arm such that the tip of the arm gets as close as possible to a predefined target point. In the first setting, the arm is placed in front of a wall whose distance away is a stochastic environment variable. At some distances, some joint angles yield a collision with the wall, which incurs a large cost. Minimising cost entails getting as close to the target as possible while avoiding the region where the wall may be present.

Figure 3a shows the expected cost (lower is better) of the arm configurations after each timestep for each method. ALOQ and random quadrature greatly outperform the other methods, including WSN. Furthermore, in this setting one-step ALOQ fails to converge at all. Figure 3b shows the learned arm configurations, as well as the policy that would be learned by ALOQ if there was no wall (No Wall). The shaded region represents the possible locations of the wall. This plot illustrates that ALOQ learns a policy that gets closest to the target. Furthermore, while all the algorithms learn to avoid the wall, active selection of θ allows ALOQ to do so more quickly: smart quadrature allows it to more efficiently observe rare events and accurately estimate their boundary. Runtime plots can be found in the supplementary material.

Next, we consider a variation in which, instead of a wall, some settings of the first joint carries a 5% probability of it breaking, which consequently incurs a large cost. Minimising cost thus entails getting as close to the target as possible, while minimising the probability of the joint breaking.

Figure 4a shows the expected cost (lower is better) of the arm configurations after each timestep for each method. We could not run WSN since θ is continuous in this setting. ALOQ converges much faster than the other baselines. Figure 4b shows the learned arm configurations together with the policy that would be learned if there were no SREs (‘No break’). The shaded region represents the joint angles that can lead to failure. This figure illustrates that ALOQ learns a qualitatively different policy than the other algorithms, one which avoids the joint angles that might lead to a breakage while still getting close to the target faster than the other methods.

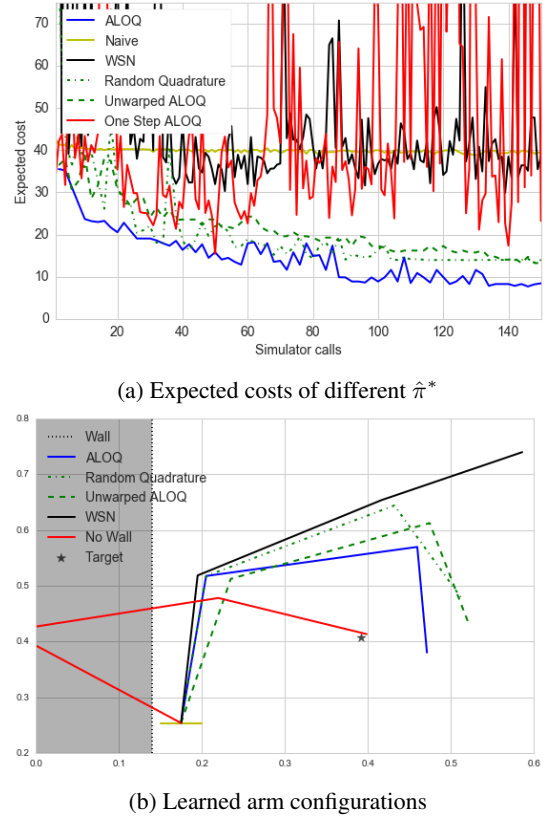
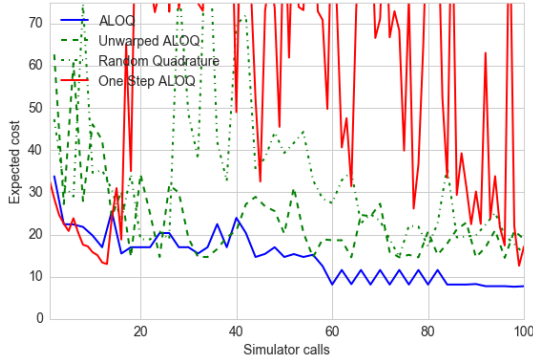


Figure 3: Performance and learned configurations of each method on the robotic arm wall collision setting.

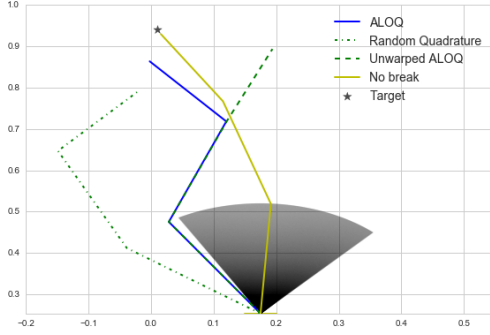
Model-Based Reinforcement Learning

Finally, we apply ALOQ to a model-based RL setting. Suppose that we do not have an accurate simulator, i.e., model, a priori but must learn one from observed physical trials. Given a decent baseline policy, we can generate trajectories in the environment and use them to compute a MAP estimate of the unknown model parameters. We could then learn an optimal policy for the resulting model. However, doing so ignores model uncertainty.

A better approach is to compute a full posterior distribution over the model parameters that captures uncertainty



(a) Expected costs of different $\hat{\pi}^*$



(b) Learned arm configurations

Figure 4: Performance and learned configurations of each method on the robotic arm breakage setting.

about, e.g., how the environment behaves under policies that are different from the baseline. We can then learn a policy that is optimal in expectation with respect to this posterior. However, finding such a policy is difficult if there are low-probability models that substantially affect expected performance, e.g., unlikely models in which an otherwise good policy will cause an arm to break.

Fortunately, ALOQ is ideally suited to this problem. The main idea is simply to treat the model parameters as environment variables. Applying ALOQ then ensures that the learned policy is robust to low-probability models, thereby allowing safe deployment of the policy in the environment. Note that ALOQ is useful for model-based RL even when the task itself contains no SREs, as long as there are important models that are unlikely according to the posterior.

To demonstrate this experimentally, we consider a setting where, instead of directly setting the robot arm’s joint angles, we set the torque applied to each joint (π). The final joint angles are determined by the torque and the unknown friction between the joints (θ). Setting the torque too high can lead to the joint breaking, which incurs a large cost.

We use the simulator as a proxy for both real trials as well as the simulated trials. In the first case, we simply sample θ from a prior, run a baseline policy, and use the observed returns to compute an approximate posterior over θ . We then use ALOQ to compute the optimal policy over this posterior (‘ALOQ policy’). For comparison, we also compute the MAP of θ and the corresponding optimal policy (‘MAP pol-

icy’). To show that active selection of θ is advantageous, we also compare against the policy learned by random quadrature. Note that PILCO (Deisenroth and Rasmussen 2011), another GP-based method that addresses model uncertainty, cannot be applied directly in this case since it requires the returns to follow certain functional forms that do not admit easy modelling of SREs.

Since we are approximating the true posterior with a set of samples, it makes sense to keep the sample size relatively low for computational efficiency when finding the ALOQ policy. However, to show that ALOQ is robust to this approximation, when comparing the performance of the ALOQ and MAP policies, we use a much larger sample for the posterior distribution.

For evaluation, we sampled θ from this more granular posterior distribution and measured the difference in the performance of the ALOQ and MAP policies. Across 20 random starts and 1000 samples of θ from the posterior, we found that on average the return of the ALOQ policy is 31% higher than the MAP policy. Figure 5 presents a histogram of the average difference across the 1000 samples. The ALOQ policy tends to slightly underperform the MAP policy in a large number of cases, but occasionally significantly outperforms it because it takes into account the unlikely models that have significant events (i.e. significantly different returns) associated with them. The ALOQ policy also performed better than random quadrature, with returns being 24% higher on average (see the histogram in the supplementary materials).

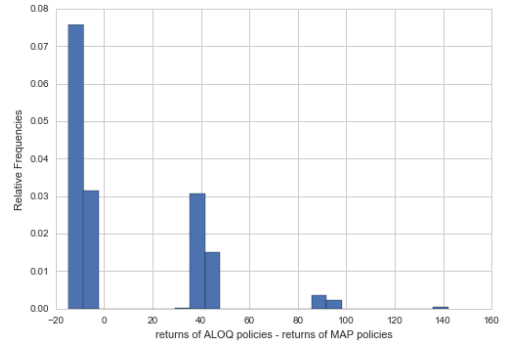


Figure 5: Comparison of the performance of the ALOQ and MAP policies under a model uncertainty scenario.

Conclusions

This paper proposed ALOQ, a novel approach to using Bayesian optimisation and quadrature to perform sample-efficient reinforcement learning in a way that is robust to the presence of significant rare events. We empirically evaluated ALOQ on test functions and a robotic arm simulator, and showed how it can be also be applied to settings involving model uncertainty. Our results demonstrated that ALOQ outperforms the WSN algorithm, which tries to address a similar problem. Further, ALOQ is more computationally efficient and is also able to handle continuous environment variables.

Acknowledgments We would like to thank Jean-Baptiste Mouret and Federico Allocati for providing the robotic arm simulator used in the experiments, and Tom Rainforth and Yannis M. Assael for the many helpful discussions. This project has received funding from the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation programme (grant agreement #637713).

References

- [Bartz-Beielstein, Lasarczyk, and Preuss 2005] Bartz-Beielstein, T.; Lasarczyk, C. W. G.; and Preuss, M. 2005. Sequential parameter optimization. In *2005 IEEE Congress on Evolutionary Computation*, volume 1, 773–780 Vol.1.
- [Briol et al. 2015] Briol, F.-X.; Oates, C. J.; Girolami, M.; Osborne, M. A.; and Sejdinovic, D. 2015. Probabilistic Integration: A Role for Statisticians in Numerical Analysis? *arXiv:1512.00933*.
- [Brochu, Cora, and de Freitas 2010] Brochu, E.; Cora, V. M.; and de Freitas, N. 2010. A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. eprint arXiv:1012.2599, arXiv.org.
- [Calandra et al. 2015] Calandra, R.; Seyfarth, A.; Peters, J.; and Deisenroth, M. 2015. Bayesian optimization for learning gaits under uncertainty. *Annals of Mathematics and Artificial Intelligence*.
- [Cox and John 1992] Cox, D. D., and John, S. 1992. A statistical method for global optimization. In *Systems, Man and Cybernetics, 1992., IEEE International Conference on*.
- [Cox and John 1997] Cox, D. D., and John, S. 1997. Sdo: A statistical method for global optimization. In *in Multidisciplinary Design Optimization: State-of-the-Art*.
- [Cully et al. 2015] Cully, A.; Clune, J.; Tarapore, D.; and Mouret, J.-B. 2015. Robots that can adapt like animals. *Nature* 521.
- [Deisenroth and Rasmussen 2011] Deisenroth, M. P., and Rasmussen, C. E. 2011. Pilco: A model-based and data-efficient approach to policy search. In *In Proceedings of the International Conference on Machine Learning*.
- [Frank, Mannor, and Precup 2008] Frank, J.; Mannor, S.; and Precup, D. 2008. Reinforcement learning in the presence of rare events. In *Proceedings of the 25th international conference on Machine learning*. ACM.
- [Hutter et al. 2009] Hutter, F.; Hoos, H. H.; Leyton-Brown, K.; and Murphy, K. P. 2009. An experimental investigation of model-based parameter optimisation: Spo and beyond. In *Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation, GECCO ’09*, 271–278. New York, NY, USA: ACM.
- [Jones, Schonlau, and Welch 1998] Jones, D.; Schonlau, M.; and Welch, W. 1998. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization* 13(4).
- [Koppejan and Whiteson 2011] Koppejan, R., and Whiteson, S. 2011. Neuroevolutionary reinforcement learning for generalized control of simulated helicopters. *Evolutionary Intelligence* 4.
- [Krause and Ong 2011] Krause, A., and Ong, C. S. 2011. Contextual gaussian process bandit optimization. In *Advances in Neural Information Processing Systems*.
- [Lizotte et al. 2007] Lizotte, D. J.; Wang, T.; Bowling, M.; and Schuurmans, D. 2007. Automatic gait optimization with gaussian process regression. In *IJCAI*.
- [Martinez-Cantin et al. 2007] Martinez-Cantin, R.; de Freitas, N.; Doucet, A.; and Castellanos, J. 2007. Active policy learning for robot planning and exploration under uncertainty. In *Robotics: Science and Systems*.
- [Martinez-Cantin et al. 2009] Martinez-Cantin, R.; de Freitas, N.; Brochu, E.; Castellanos, J.; and Doucet, A. 2009. A bayesian exploration-exploitation approach for optimal online sensing and planning with a visually guided mobile robot. *Autonomous Robots* 27(2).
- [Moćkus 1975] Moćkus, J. 1975. On bayesian methods for seeking the extremum. In *Optimization Techniques IFIP Technical Conference Novosibirsk, July 1–7, 1974*.
- [O’Hagan 1991] O’Hagan, A. 1991. Bayeshermite quadrature. *Journal of Statistical Planning and Inference* 29(3).
- [Osborne et al. 2012] Osborne, M.; Garnett, R.; Ghahramani, Z.; Duvenaud, D. K.; Roberts, S. J.; and Rasmussen, C. E. 2012. Active learning of model evidence using bayesian quadrature. In *Advances in Neural Information Processing Systems*.
- [Rasmussen and Ghahramani 2003] Rasmussen, C. E., and Ghahramani, Z. 2003. Bayesian monte carlo. *Advances in neural information processing systems* 15.
- [Rasmussen and Williams 2005] Rasmussen, C. E., and Williams, C. K. I. 2005. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press.
- [Snoek et al. 2013] Snoek, J.; Swersky, K.; Zemel, R. S.; and Adams, R. P. 2013. Input warping for bayesian optimization of non-stationary functions. *Advances in Neural Information Processing Systems Workshop on Bayesian Optimization*.
- [Williams, Santner, and Notz 2000] Williams, B. J.; Santner, T. J.; and Notz, W. I. 2000. Sequential design of computer experiments to minimize integrated response functions. *Statistica Sinica* 10(4).